

A simpler proof for $O(\text{congestion} + \text{dilation})$ packet routing

Thomas Rothvoß*

M.I.T., Cambridge, USA

Abstract

In the *store-and-forward routing* problem, packets have to be routed along given paths such that the arrival time of the latest packet is minimized. A groundbreaking result of Leighton, Maggs and Rao says that this can always be done in time $O(\text{congestion} + \text{dilation})$, where the *congestion* is the maximum number of paths using an edge and the *dilation* is the maximum length of a path. However, the analysis is quite arcane and complicated and works by iteratively improving an infeasible schedule. Here, we provide a more accessible analysis which is based on conditional expectations. Like [LMR94], our easier analysis also guarantees that constant size edge buffers suffice.

Moreover, it was an open problem stated e.g. by Wiese [Wie11], whether there is any instance where all schedules need at least $(1 + \varepsilon) \cdot (\text{congestion} + \text{dilation})$ steps, for a constant $\varepsilon > 0$. We answer this question affirmatively by making use of a probabilistic construction.

1 Introduction

One of the fundamental problems in parallel and distributed systems is to transport packets within a communication network in a timely manner. Any routing protocol has to make two kinds of decisions: (1) on which paths shall the packets be sent and (2) according to which priority rule should packets be routed along those paths, considering that communication links have usually a limited bandwidth. In this paper, we focus on the second part of the decision process. More concretely, we assume that a network in form of a directed graph $G = (V, E)$ is given, together with source sink pairs $s_i, t_i \in V$ for $i = 1, \dots, k$ and s_i - t_i paths $P_i \subseteq E$. So the goal is to route the packets from their source along the given path to their sink in such a way that the *makespan* is minimized. Here, the *makespan* denotes the time when the last packet arrives at its destination. Moreover, we assume *unit bandwidth* and *unit transit time*, i.e. in each time unit only one packet can traverse an edge and the traversal takes exactly one time unit. Since the only freedom for the scheduler lies in the decision when packets move and when they wait, this setting is usually called *store and forward routing*. Note that we make no assumption about the structure of the graph or the paths. In fact, we can allow that the graph has multi-edges and loops; a path may even revisit the same node several times. We only forbid that a path uses the same edge more than once.

Two natural parameters of the instance are the *congestion* $C := \max_{e \in E} |\{i \mid e \in P_i\}|$, i.e. the maximum number of paths that share a common edge and the *dilation* $D := \max_{i=1, \dots, k} |P_i|$, i.e. the length of the longest path. Obviously, for any instance, both parameters C and D are lower bounds on the makespan for any possible routing policy. Surprisingly, Leighton, Maggs and Rao [LMR94] could prove that the optimum achievable makespan is always within a constant factor of $C + D$. Since then,

*Email: rothvoss@math.mit.edu. Supported by the Alexander von Humboldt Foundation within the Feodor Lynen program, by ONR grant N00014-11-1-0053 and by NSF contract CCF-0829878.

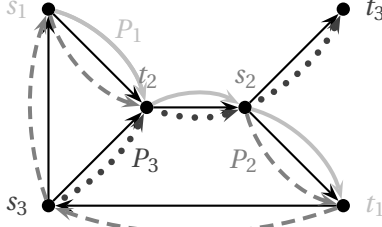


Figure 1: Example instance with $k = 3$, $C = 2$ and $D = 4$

their approach has been revisited several times. First, [LMR99] provided a polynomial time algorithm that makes the approach constructive (which nowadays would be easy using the Moser Tardos algorithm [MT10]). Scheideler [Sch98, Chapter 6] provides a more careful (and more accessible) analysis which reduces the hidden constants to $39(C + D)$. More recently Peis and Wiese [PW11] reduced the constant to 24 (and beyond, for larger minimum bandwidth or transit time).

Already the original paper of [LMR94] also showed that (huge) constant size *edge buffers* are sufficient. Scheideler [Sch98] proved that even a buffer size of 2 is enough. However, all proofs [LMR94, LMR99, Sch98, PW11] use the original idea of Leighton, Maggs and Rao to start with an infeasible schedule and insert iteratively random delays to reduce the infeasibility until no more than $O(1)$ packets use an edge per time step (in each iteration, applying the Lovász Local Lemma).

In this paper, we suggest a somewhat dual approach in which we start with a probabilistic schedule which is feasible in expectation and then reduce step by step the randomness (still making use of the Local Lemma). Our construction here is not fundamentally different from the original work of [LMR94], but the emerging proof is “less iterative” and, in the opinion of the author, also more clear and explicit in demonstrating to the reader *why* a constant factor suffices. Especially obtaining the additional property of constant size edge buffers is fairly simple in our construction.

If it comes to lower bounds for general routing strategies, the following instance is essentially the worst known one: C many packets share the same path of length D . Then it takes C time units until the last packet crosses the first edge; that packet needs $D - 1$ more time units to reach its destination, leading to a makespan of $C + D - 1$. Wiese [Wie11] states that no example is known where the optimum makespan needs to be even a small constant factor larger. We answer the open question in [Wie11] and show that for a universal constant $\varepsilon > 0$, there is a family of instances in which every routing policy needs at least $(1 + \varepsilon) \cdot (C + D)$ time units (and $C, D \rightarrow \infty$)¹. In our chosen instance, we generate paths from random permutations and use probabilistic arguments for the analysis.

1.1 Related Work

The result of [LMR94, LMR99] could be interpreted as a constant factor approximation algorithm for the problem of finding the minimum makespan. In contrast, finding the optimum schedule is **NP**-hard [CI96]. In fact, even on trees, the problem remains **APX**-hard [PSW09]. If we generalize the problem to finding paths plus schedules, then constant factor approximation algorithms are still possible due to Srinivasan and Teo [ST00] (using the fact that it suffices to find paths that minimize the sum of congestion and dilation). Koch et al. [KPSW09] extend this to a more general setting, where messages consisting of several packets have to be sent.

¹The constant can be chosen e.g. as $\varepsilon := 0.00001$, though we do not make any attempt to optimize the constant, but focus on a simple exposition.

The Leighton-Maggs-Rao result, apart from being quite involved, has the disadvantage of being a non-local offline algorithm. In contrast, there is a distributed algorithm with makespan $O(C) + (\log^* n)^{O(\log^* n)} D + \log^{O(1)} n$ by Rabani and Tardos [RT96] which was later improved to $O(C + D + \log^{1+\varepsilon} n)$ by Ostrovsky and Rabani [OR97]. If the paths are indeed shortest paths, then there is a randomized online routing policy which finishes in $O(C + D + \log k)$ steps [MV99]. To the best of our knowledge, the question concerning the existence of an $O(C + D)$ online algorithm is still open. We refer to the book of Scheideler [Sch98] for a more detailed overview about routing policies.

One can also reinterpret the packet routing problem as *(acyclic) job shop scheduling* $J \mid p_{ij} = 1, \text{acyclic} \mid C_{\max}$, where jobs J and machines M are given. Each job has a sequence of machines that it needs to be processed on in a given order (each machine appears at most once in this sequence), while all processing times have unit length. For the natural generalization $(J \mid p_{ij}, \text{acyclic} \mid C_{\max})$ with arbitrary processing times p_{ij} , Feige & Scheideler [FS02] showed that schedules of length $O(L \cdot \log L \cdot \log \log L)$ are always possible and for some instances, every schedule needs at least $\Omega(L \cdot \frac{\log L}{\log \log L})$ time units, where we abbreviate $L := \max\{C, D\}$.² Svensson and Mastrolilli [MS11] showed that this lower bound even holds in the special case of *flow shop scheduling*, where all jobs need to be processed on all machines in the same order (in packet routing, this corresponds to the case that all paths P_i are identical). In fact, for *flow shop scheduling with jumps* (i.e. each job needs to be processed on a given subset of machines) it is even **NP**-hard to approximate the optimum makespan within any constant factor [MS11].

In contrast, if we allow preemption, then even for acyclic job shop scheduling, the makespan can be reduced to $O(C + D \log \log \max_{ij} p_{ij})$ [FS02] and it is conceivable that even $O(C + D)$ might suffice.

1.2 Organisation

In Section 2, we recall some probabilistic tools. Then in Section 3 we show the existence of an $O(C + D)$ routing policy, which is modified in Section 4 to guarantee that constant size edge buffers suffice. Finally, we show the lower bound in Section 5.

2 Preliminaries

Later, we will need the following concentration result, which is a version of the *Chernov-Hoeffding bound*:

Lemma 1 ([DP09, Theorem 1.1]). *Let $Z_1, \dots, Z_k \in [0, \delta]$ be independently distributed random variables with sum $Z := \sum_{i=1}^k Z_i$ and let $\mu \geq \mathbb{E}[Z]$. Then for any $\varepsilon > 0$,*

$$\Pr[Z > (1 + \varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{3} \cdot \frac{\mu}{\delta}\right).$$

Moreover, we need the *Lovász Local Lemma* (see also the books [AS08] and [MU05] and for the constructive version, see [MT10]).

Lemma 2 (Lovász Local Lemma [EL75]). *Let A_1, \dots, A_m be arbitrary events such that (1) $\Pr[A_i] \leq p$; (2) each A_i depends on at most d many other events; and (3) $4 \cdot p \cdot d \leq 1$. Then $\Pr[\bigcap_{i=1}^m \bar{A}_i] > 0$.*

²In this setting, one extends $C = \max_{i \in M} \sum_{j \in J: j \text{ uses } i} p_{ij}$ and $D = \max_{j \in J} \sum_{i \in M: j \text{ uses } i} p_{ij}$.

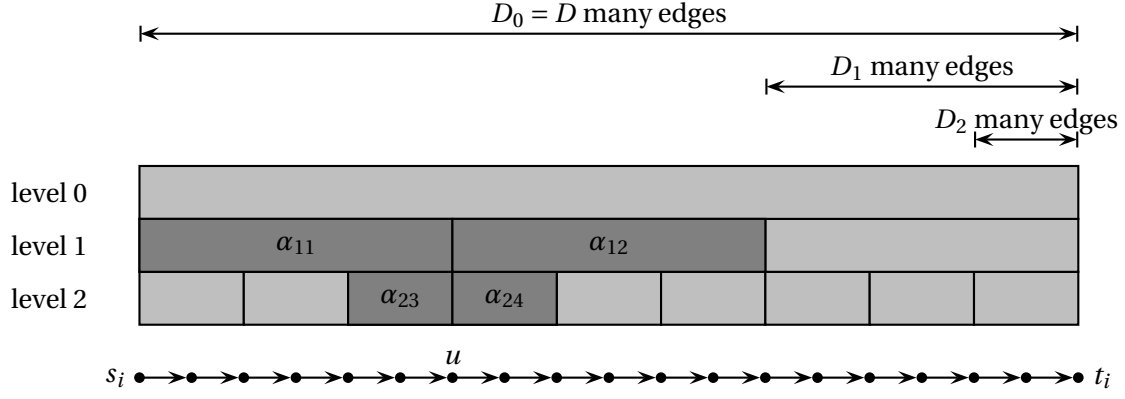


Figure 2: Path P_i and its dissection with $L = 2$. Denote the random waiting time of the j th block in level ℓ by $\alpha_{\ell j}$. Then the packet would wait for $(W_1 - \alpha_{11}) + \alpha_{12} + (W_2 - \alpha_{23}) + \alpha_{24}$ time units in node u .

3 $O(\text{congestion} + \text{dilation})$ routing

After adding dummy paths and edges, we may assume that $C = D$ and every path has length exactly D . In the following we show how to route the packets within $O(D)$ time units such that in each time step, each edge is traversed by at most $O(1)$ many packets (by stretching the time by another $O(1)$ factor, one can obtain a schedule with makespan $O(D)$ in which each edge is indeed only traversed by a single packet). In the following, we call the largest number of packets that traverse the same edge in one time unit the *load* of the schedule.

Let $\Delta > 0$ be a constant that we leave undetermined for now – at several places we will simply assume Δ to be large enough for our purpose. Consider a packet i and partition its path P_i into a laminar family of *blocks* such that the blocks on level ℓ contain $D_\ell = D^{(1/2)^\ell}$ many consecutive edges.³ We stop this dissection, when the last block (whose index we denote by L) has length between Δ and Δ^2 .

In other words, the root block (i.e. the path P_i itself) is on level 0 and the depth of that laminar family is $L = \Theta(\log \log D)$ (though this quantity will be irrelevant for the analysis). Each block has 2 *boundary nodes*, a *start node* and an *end node*. Observe that a level ℓ block of length D_ℓ has children of length $D_{\ell+1} = \sqrt{D_\ell}$. Moreover, we define

$$W_\ell := \begin{cases} D_\ell & \ell = 0 \\ D_\ell^{1/4} & \ell \geq 1 \end{cases}$$

The routing policy for packet i is now as follows: For each level ℓ block, the packet waits a uniformly and independently chosen random time $x \in [1, W_\ell]$ at the start node⁴; furthermore the packet waits $W_\ell - x$ time units at the end node (see Figure 2). This policy has two crucial properties:

- (A) The total waiting time of each packet is $O(D)$.
- (B) The time t at which packet i crosses an edge $e \in P_i$ is a random variable that depends only on the random waiting times of the blocks that contain e — in fact, i.e. only one block from each level.

³Depending on D , the quantity D_ℓ may not be integral. But all our calculations have enough slack so that one could replace D_ℓ with the nearest power of 2. Then we may also assume that for each ℓ , D_ℓ divides $D_{\ell-1}$.

⁴We define $[a, b] := \{a, a+1, a+2, \dots, b\}$ as the set of integers between a and b .

Let us argue, why (A) is true. The waiting time on level $\ell = 0$ will be precisely D , while for each $\ell \geq 1$ the total level- ℓ waiting time for each packet will be $\frac{D}{D_\ell} \cdot W_\ell = \frac{D}{D_\ell^{3/4}}$. Using the crude bound $D_\ell \geq 4 \cdot D_{\ell+1}$ we have $D_{L-j} \geq 4^j$, hence on level $L-j > 0$, the total waiting time will be at most $\frac{D}{D_{L-j}^{3/4}} \leq \frac{D}{2^j}$. Thus the total waiting time for a packet, summed over all levels is at most $D + D \sum_{j=0}^{L-1} (\frac{1}{2})^j = O(D)$. In other words: each packet is guaranteed to arrive after at most $T := O(D)$ time units. Note that there are instances where the vast majority of random outcomes would yield a superconstant load on some edge. However, one can prove that there *exists* a choice of the waiting times such that the load does not exceed $O(1)$.

Let $X(e, t, i) \in \{0, 1\}$ be the random variable that tells us whether packet i is crossing edge e at time t . Moreover, let $X(e, t) = \sum_{i=1}^k X(e, t, i)$ be the number of packets crossing e at time t . Since packet i waits a random time from $[1, D]$ in s_i , we have $\Pr[X(e, t, i)] \leq \frac{1}{D}$ for each e, i, t (more formally: no matter how the waiting times on level ≥ 1 are chosen, there is always at most one out of D outcomes for the level 0 waiting time that cause packet i to cross e precisely at time t). Since no edge is contained in more than D paths, we have $\mathbb{E}[X(e, t)] \leq 1$.

In the following, if $\alpha \in [1, W_\ell]^{D/D_\ell}$ is a vector of level ℓ -waiting times, then $\mathbb{E}[X(e, t) \mid \alpha]$ denotes the corresponding conditional expectation, depending on α . The idea for the analysis is to fix the waiting times on one level at a time (starting with level 0) such that the conditional expectation $\mathbb{E}[X(e, t)]$ never increases to a value larger than, say 2. Before we continue, we want to be clear about the behaviour of such conditional random variables.

Lemma 3. *Let $\ell \in \{0, \dots, L-1\}$ and condition on arbitrary waiting times for level $0, \dots, \ell$. Then for any packet i , edge $e \in E$ and any time $t \in [T]$ one has*

- a) $\Pr[X(e, t, i)] \leq \frac{1}{W_{\ell+1}}$.
- b) *If the event $X(e, t, i)$ has non-zero probability, then $\Pr[X(e, t, i)] \geq \frac{1}{W_{\ell+1}^2}$.*

Proof. For (a), suppose also all waiting times except of the level $\ell+1$ block in which i crosses e are fixed adversarially. Still, there is at most one out of $W_{\ell+1}$ outcomes that cause packet i to cross e at time t .

For (b), observe that the time at which packet i crosses e depends only on the waiting time of the blocks that contain e (i.e. one block per level). The number of possible outcomes of those waiting times is bounded by $\prod_{j=0}^{L-\ell-1} W_{\ell+1+j} \leq (W_{\ell+1})^{\sum_{j \geq 0} (1/2)^j} = W_{\ell+1}^2$. \square

The whole analysis boils down to the following lemma, in which we prove that we can always fix the waiting times on level ℓ without increasing the expected load on any edge by more than $D_\ell^{-1/32}$. What happens formally is that we show the existence of a sequence $\alpha_0, \dots, \alpha_{L-1}$ such that α_ℓ denotes a vector of level ℓ -waiting times and

$$\mathbb{E}[X(e, t) \mid \alpha_0, \dots, \alpha_{\ell-1}, \alpha_\ell] \leq \mathbb{E}[X(e, t) \mid \alpha_0, \dots, \alpha_{\ell-1}] + \frac{1}{D_\ell^{1/32}} \quad \forall e \in E \forall t \in [T] \quad (1)$$

(given that the right hand side is at least 1). To do this, suppose we already found and fixed proper waiting times $\alpha_0, \dots, \alpha_{\ell-1}$. Then one can interpret the left hand side of (1) as a random variable depending on α_ℓ , which is the sum of independently distributed values — and hence well concentrated. Moreover the dependence degree of this random variable is bounded by a polynomial in D_ℓ . Thus the Lovász Local Lemma provides the existence of suitable waiting times α_ℓ .

Lemma 4. Let $\ell \in \{0, \dots, L-1\}$ and suppose that we already fixed all waiting times on level $0, \dots, \ell-1$. Let $X(e, t)$ be the corresponding conditional random variable and assume $\gamma \geq \max_{e \in E, t \in [T]} \{\mathbb{E}[X(e, t)]\}$ and $1 \leq \gamma \leq 2$. Then there are level ℓ waiting times α such that

$$\mathbb{E}[X(e, t) \mid \alpha] \leq \gamma + \frac{1}{D_\ell^{1/32}} \quad \forall e \in E \forall t \in [T]$$

Proof. We abbreviate $m := D_\ell$. First recall that on level ℓ , (1) blocks have length m ; (2) the child blocks have length \sqrt{m} and (3) the waiting time on the next level $\ell+1$ is from $[1, m^{1/8}]$.

We define $Y(e, t) := \mathbb{E}[X(e, t) \mid \alpha]$ and consider $Y(e, t)$ as a random variable only depending on α . Since the waiting times on levels $0, \dots, \ell-1$ are already fixed, we know exactly the level ℓ -block in which packet i will cross edge e — let $\alpha_{i,e}$ be the random waiting time for that block. Then we can write

$$Y(e, t) = \sum_{i=1}^k \Pr[X(e, t, i) \mid \alpha_{i,e}] \quad (2)$$

By Lemma 3.(b), we know that $\Pr[X(e, t, i) \mid \alpha_{i,e}] \leq \frac{1}{m^{1/8}}$ for every choice of $\alpha_{i,e}$. Thus $Y(e, t)$ is the sum of independent random variables in the interval $[0, m^{-1/8}]$ and the Chernov bound (Lemma 1) provides

$$\Pr\left[Y(e, t) > \gamma + \frac{1}{m^{1/32}}\right] \leq \exp\left(-\frac{1}{3} \cdot \frac{1}{(2m^{1/32})^2} \cdot m^{1/8}\right) \leq e^{-m^{1/16}/12}$$

Now we want to apply the Lovász Local Lemma for the events “ $Y(e, t) > \gamma + m^{-1/32}$ ” to argue that it is possible that none of the events happens. So it suffices to bound the dependence degree by a polynomial in m . Lemma 3.(b) guarantees that if the event $X(e, t, i)$ is possible at all, then $\Pr[X(e, t, i)] \geq \frac{1}{W_\ell^2} \geq \frac{1}{m}$. Now, reconsider Equation (2) and let $Q(e, t) := \{i \in [k] \mid \Pr[X(e, t, i)] > 0\}$ be the set of packets that still have a non-zero chance to cross edge e at time t . Taking expectations of Equation (2), we see that

$$2 \geq \gamma \geq \mathbb{E}[Y(e, t)] = \sum_{i \in Q(e, t)} \Pr[X(e, t, i)] \geq |Q(e, t)| \cdot \frac{1}{m}$$

and hence $|Q(e, t)| \leq 2m$. This means that each random variable $Y(e, t)$ depends on at most $2m$ entries of α . Moreover, consider an entry in α , say it belongs to packet i and block B . This random variable appears in the definition of $Y(e, t)$ if $e \in B$ and t belongs to B 's time frame — these are just $m \cdot O(m)$ many combinations. Here we use that the time difference between entering a level ℓ block and leaving it, is bounded by $O(D_\ell)$. Overall, the dependence degree is $O(m^3)$. Since the probability of each bad event “ $Y(e, t) > \gamma + m^{-1/32}$ ” is superpolynomially small, the claim follows by the Lovász Local Lemma and the assumption that $m \geq \Delta$ is large enough. \square

We apply this lemma for $\ell = 0, \dots, L-1$ and the maximum load after any iteration will be bounded by $1 + \sum_{\ell=0}^{L-1} (D_\ell)^{-1/32} \leq 2$ for Δ large enough. The finally obtained random variables $X(e, t, i)$ are *almost* deterministic — just the waiting times on level L are still probabilistic. But again by Lemma 3, all non-zero probabilities $\Pr[X(e, t, i)]$ are at least $\frac{1}{(\Delta^{1/4})^2} = \Omega(1)$, thus making an arbitrary choice for them cannot increase the load by more than a constant factor. Finally, we end up with a schedule with load $O(1)$.

4 Providing constant size edge buffers

Now let us imagine that each directed edge $(u, v) \in E$ has an *edge buffer* at the beginning of the edge. Whenever a packet arrives at node u and has e as next edge on its path, the packet waits in e 's edge

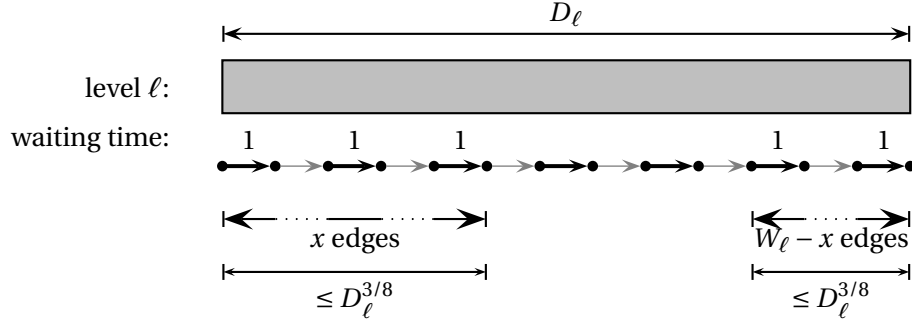


Figure 3: Visualisation of the waiting time for a single level ℓ -block. Black edges are assigned to the considered block and are labelled with the waiting time.

buffer. But a packet i is still allowed to wait an arbitrary amount of time in s_i or t_i .

In the construction that we saw above, it may happen that many packets wait for a long time in one node, i.e. a large edge buffer might be needed. However, as was shown by Leighton, Maggs and Rao [LMR94], one can find a schedule such that edge buffers of size $O(1)$ suffice. More precisely, [LMR94] found a schedule with load $O(1)$ in which each packet waits at most one time unit in every node — after stretching, this results in a schedule with load 1 and $O(1)$ buffer size.

In fact, we can modify the construction from Section 3 in such a way that we *spread* the waiting time over several edges and obtain the same property. Consider the dissection from the last section. Iteratively, for $\ell = 1, \dots, L$, shift the level ℓ -blocks such that every level $\ell - 1$ boundary node lies in the middle of some level ℓ -block, see Figure 4 (note that we assume that $D_{\ell-1}$ is an integral multiple of D_ℓ). Fix a packet i and denote the edges of its path by $P_i = (e_1, \dots, e_D)$, then we assign all edges e_j whose index j is of the form $(1 + 2\mathbb{Z}) \cdot 2^q$ to level $L - q$ (for $q \in \{0, \dots, L - 1\}$). For example, this means that all odd edges are assigned to the last level; the top level does not get assigned any edges.

Now we again define random waiting times for packet i and a block B : on level $\ell \geq 1$, each block picks a uniform random number $x \in [1, W_\ell]$. The packet waits on each of the first x edges that are assigned to the block. Moreover, it waits on each of the last $W_\ell - x$ edges that are assigned to the block. Observe that regardless of the random outcome, the packet will wait at most once per edge since edges are assigned to at most one level. Using the convenient bound $2^{L-\ell} \leq D_\ell^{1/8}$ for Δ large enough, we see that all level- ℓ randomization takes place within the first and last $D_\ell^{3/8}$ edges of each block, see Figure 3.

The top block does not get assigned any edge, so instead for each packet i , we pick a value $x \in [1, D]$ at random and wait x time units in s_i .⁵

Reinspecting Lemma 3, we observe that Lemma 3.b) holds without any alterations and Lemma 3.a) holds as long as the considered edge e has a minimum distance of $D_{\ell+1}^{3/8}$ from the nearest level $\ell + 1$ boundary node. Surprisingly, also Lemma 4 still holds with a minor modification in the claimed bound.

Lemma 5. *Let $\ell \in \{0, \dots, L - 2\}$ and suppose that we already fixed all waiting times on level $0, \dots, \ell - 1$. Let $X(e, t)$ be the corresponding conditional random variables and assume $\gamma \geq \max_{e \in E, t \in [T]} \{\mathbb{E}[X(e, t)]\}$ and $1 \leq \gamma \leq 2$. Then there are level ℓ waiting times α such that*

$$\mathbb{E}[X(e, t) \mid \alpha] \leq \gamma + \frac{1}{D_\ell^{1/64}} \quad \forall e \in E \quad \forall t \in [T]$$

⁵If for a block, due to the shifting, some or all waiting edges are shifted “before” the source s_i , then the packet just waits the missing time in s_i .

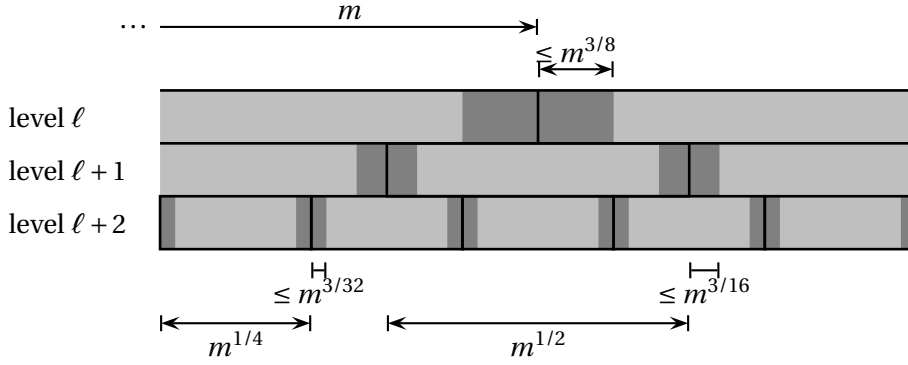


Figure 4: Shifted dissection (with $m := D_\ell$). Regions in which randomization takes place are depicted in darkgray (observe these regions do not overlap for consecutive levels).

Proof. Again abbreviate $m := D_\ell$ and consider

$$Y(e, t) := \mathbb{E}[X(e, t) \mid \alpha] = \sum_{i=1}^k \Pr[X(e, t, i) \mid \alpha_{i,e}]$$

as a random variable only depending on α (recall that $\alpha_{i,e}$ is the random waiting time for that level ℓ -block in which packet i crosses edge e). For a fixed edge e , for one of those levels $\ell' \in \{\ell+1, \ell+2\}$, the edge e is at least $\frac{1}{4}m^{1/4}$ edges away from the next level ℓ' boundary node (see Figure 4). Consider the level ℓ' -block B that contains e . As already argued, all randomization takes place on the first and last $D_{\ell'}^{3/8} \leq D_{\ell+1}^{3/8} = m^{3/16} \ll \frac{1}{4}m^{1/4}$ edges (for $m \geq \Delta$ large enough). So we can still apply Lemma 3.a) for level ℓ' to obtain $\Pr[X(e, t, i) \mid \alpha_{i,e}] \leq \frac{1}{W_{\ell'}} \leq \frac{1}{m^{1/16}}$. Again by the Chernov bound (i.e. Lemma 1 with $\delta := \frac{1}{m^{1/16}}$, $\varepsilon := \frac{1}{2m^{1/64}}$, $\mu := \gamma \geq 1$) we have

$$\Pr\left[Y(e, t) > \gamma + \frac{1}{m^{1/64}}\right] \leq \exp\left(-\frac{1}{3} \cdot \frac{1}{(2m^{1/64})^2} \cdot m^{1/16}\right) = e^{-m^{1/32}/12}$$

Next, note that still $\Pr[X(e, t, i) \mid \alpha_{i,e}] \geq \frac{1}{m}$, given that this probability is positive. Thus from now on we can follow the arguments in the proof of Lemma 4. The dependence degree is still bounded by $O(m^3)$, thus the claim follows by the Lovász Local Lemma since $4 \cdot O(m^3) \cdot e^{-m^{1/32}/12} \leq 1$ for $m \geq \Delta$ large enough. \square

Again, we have initially $\mathbb{E}[X(e, t)] \leq 1$ for all e and t , then we fix the waiting times iteratively on level $0, \dots, L-2$ using Lemma 5 and make an arbitrary choice for the waiting times of level $L-1$ and level L . This results in a schedule of length $O(D)$ and load $O(1)$, in which packets wait at most one time unit before entering an edge.

5 A $(1 + \varepsilon) \cdot (C + D)$ lower bound

In this section, we prove that there is an instance in which the optimum makespan must be at least $(1 + \varepsilon) \cdot (C + D)$, where $\varepsilon > 0$ is a small constant. The graph $G = (V, E)$ is defined as depicted in Figure 5 (the formal definition follows from the definition of the paths, which we will see in a second). Edges $e_i = (u_i, v_i)$ are called *critical edges*, while we term (v_i, u_j) *back edges*. We want to choose paths P_1, \dots, P_n as random paths through the network, all starting at $s_i := s$ and ending at $t_i := t$. More

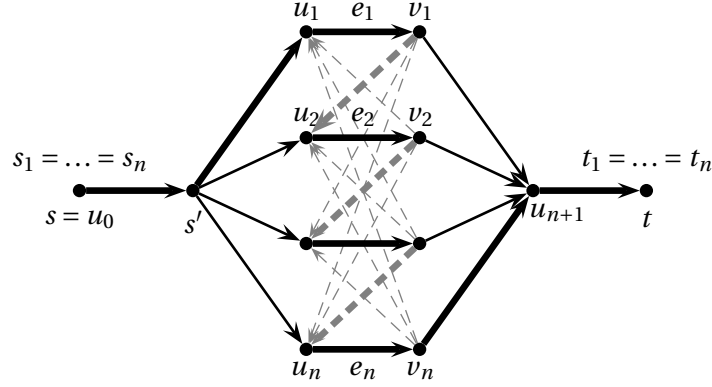


Figure 5: Graph $G = (V, E)$. The bold edges depict the path P_i originating from the identity permutation $\pi_i = (1, 2, \dots, n)$.

concretely, each packet i picks a uniform random permutation $\pi_i : [n] \rightarrow [n]$ which gives the order in which it moves through the critical edges e_1, \dots, e_n . In other words,

$$P_i = (s, s', u_{\pi_i(1)}, v_{\pi_i(1)}, u_{\pi_i(2)}, v_{\pi_i(2)}, \dots, u_{\pi_i(n)}, v_{\pi_i(n)}, u_{n+1}, t).$$

Then the congestion is n and the dilation is $2n + 3$. We consider the time frame $[1, T]$ with $T = (3 + \varepsilon)n$ and claim that for $\varepsilon > 0$ small enough, there will be no valid routing that is finished by time T .

Theorem 6. *Pick paths P_1, \dots, P_n at random. Then with probability $1 - e^{-\Omega(n^2)}$, there is no packet routing policy with makespan at most $3.000032n$ (even if buffers of unlimited size are used).*

First of all, clearly the makespan must be at least $C + D - 1 \approx 3n$ since all paths have the same length D and all packets must first cross edge (s, s') . So if we allow only time $(3 + \varepsilon)n$, then there is only a small slack of εn time units. One can show that the number of different possible routing strategies is bounded by $2^{o(n^2)}$ (for $\varepsilon \rightarrow 0$). In contrast, we can argue that a *fixed* routing will fail against random paths with probability $2^{-\Omega(n^2)}$. Then choosing ε small enough, the theorem follows using the union bound over all routing strategies.

We call a packet i *active* at time τ if it is traversing an edge. We say a packet is *parking* at time τ if it is either in the end node t_i nor in the start node s_i . We say a packet is *waiting* if it is neither active nor parking.

5.1 The number of potential routing strategies

Consider a fixed packet i and let us discuss, how a routing strategy is defined. The only decision that is made, is of the form: “*How many time units shall the packet wait in the k -th node on its path (for $k = 0, \dots, D$)*”. It is not necessary to wait in s' since a packet could instead move to $u_{\pi_i(1)}$ and wait there. Moreover, it is not needed to wait in one of the nodes v_j , since instead it could also wait in the next $u_{j'}$ node on its way (the reason is that if there would be a collision on a back edge $(v_{\pi_i(j)}, u_{\pi_i(j+1)})$ with packet $i' \neq i$, then this packet i' has crossed the critical edge $(u_{\pi_i(j)}, v_{\pi_i(j)})$ together with i in the previous time step, so there was already a collision). In other words, the complete routing strategy for packet i can be described as a $(n+2)$ -dimensional vector $W_i \in \mathbb{Z}_{\geq 0}^{n+2}$, where W_{ij} is the time that packet i stays in node u_j (for convenience, we denote s also as u_0). Then $\sum_{j=1}^{n+1} W_{ij}$ is the total waiting time and for $i \in [n]$ and W_{i0} is the time that i parks in the start node.

Independently from the outcome of the random experiment, we know the time when each packet crosses the edges incident to s and to t . We call W a *candidate routing strategy*, if there is no collision on (s, s') and (u_{n+1}, t) and the makespan of each packet is bounded by $(3 + \varepsilon)n$.

Recall that $H(\delta) = \delta \log \frac{1}{\delta} + (1 - \delta) \log \frac{1}{1 - \delta}$ is the *binary entropy function*⁶. Then we have:

Lemma 7. *The total number of candidate routing matrices W is at most $2^{(\Phi(\varepsilon) + o(1)) \cdot n^2}$, where $\Phi(\varepsilon) := H(\frac{\varepsilon}{1 + \varepsilon}) \cdot (1 + \varepsilon)$.*

Proof. First of all, the parking times in s and the total waiting time $\sum_{j=1}^{n+1} W_{ij}$ for a packet i are between 0 and $(1 + \varepsilon)n \leq 2n$; thus there are at most $(2n)^{2n} = 2^{o(n^2)}$ many possibilities to choose them.

Thus assume that the total waiting time $\varepsilon_i n = \sum_{j=1}^{n+1} W_{ij}$ for packet i is fixed. Then the number of possibilities how this waiting time can be distributed among nodes u_1, \dots, u_{n+1} is bounded by

$$\binom{(n+1) + (\varepsilon_i n) - 1}{\varepsilon_i n} \leq 2^{H(\frac{\varepsilon_i}{1 + \varepsilon_i}) \cdot (1 + \varepsilon_i) \cdot n} = 2^{\Phi(\varepsilon_i) \cdot n}$$

where we use the bound $\binom{m}{\delta m} \leq 2^{H(\delta)m}$ with $m = (1 + \varepsilon_i)n$ and $\delta = \frac{\varepsilon_i}{1 + \varepsilon_i}$.

Next, let us upperbound the total waiting time $n \sum_{i=1}^n \varepsilon_i$. Of course, the waiting time must fit into the time frame of length $T = (3 + \varepsilon)n$. Since edge (s, s') can only be crossed by one packet at a time, the cumulated time that the packets spend in the start node is at least $\sum_{\tau=0}^{n-1} \tau \approx \frac{n^2(1 - o(1))}{2}$. The same amount of time is spent by all packets in the end node. Moreover, the packets spend at least $2n^2$ time units traversing edges. We conclude that

$$n \sum_{i=1}^n \varepsilon_i \leq nT - \frac{n^2(1 - o(1))}{2} - \frac{n^2(1 - o(1))}{2} - 2n^2 = (\varepsilon + o(1))n^2,$$

thus $\sum_{i=1}^n \varepsilon_i \leq (\varepsilon + o(1))n$. Once the values $\varepsilon_1, \dots, \varepsilon_n$ are fixed, the total number of routing policies for the n packets is hence upperbounded by

$$\prod_{i=1}^n 2^{\Phi(\varepsilon_i)n} = 2^{n \sum_{i=1}^n \Phi(\varepsilon_i)} \leq 2^{n^2 \Phi(\frac{1}{n} \sum_{i=1}^n \varepsilon_i)} \leq 2^{n^2 (\Phi(\varepsilon) + o(1))}$$

Here we use Jensen's inequality together with the fact that Φ is concave. The claim follows. \square

The important property of function Φ apart from concavity is that $\lim_{\varepsilon \rightarrow 0} \Phi(\varepsilon) = 0$. Note that for $0 \leq \varepsilon \leq \frac{1}{10}$, one can conveniently upperbound $\Phi(\varepsilon) \leq 2^{1.5\varepsilon \log(\frac{1}{\varepsilon})n}$.

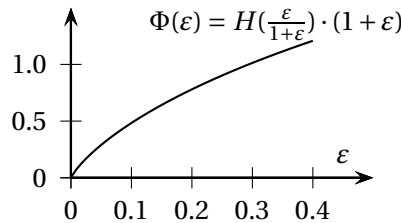


Figure 6: Function Φ .

⁶Here log is the binary logarithm.

5.2 A fixed strategy vs. random paths

Now consider a *fixed* candidate routing matrix W and imagine that the paths are taken at random. We will show that this particular routing matrix W is not legal with probability $1 - e^{-\Omega(n^2)}$. For this sake, we observe that there must be $\Omega(n)$ time units in which at least a constant fraction of packets cross critical edges. For each such time unit the probability of having no collision is at most $(\frac{1}{2})^{\Omega(n)}$ and the claim follows. The only technical difficulty lies in the fact that the outcomes of values $\pi_i(j)$ and $\pi_i(j')$ for the random permutations are (mildly) dependent.

Lemma 8. *Suppose $\varepsilon \leq \frac{1}{20}$. Let W be a candidate routing matrix. Then take paths P_1, \dots, P_n at random. The probability that the routing scheme defined by W is collision-free is at most $(\frac{15}{16})^{n^2/128}$.*

Proof. For time τ , let $\beta_\tau n$ be the number of packets that cross one of the critical edges at time τ , thus $\sum_{\tau=1}^T \beta_\tau = n$ (note that the β_τ 's do not depend on the random experiment). Let $p := \Pr_{\tau \in [T]} [\beta_\tau \geq \frac{1}{4}]$ be the fraction of time units in which at least $\frac{n}{4}$ packets are crossing a critical edge. Then

$$\frac{1}{3 + \varepsilon} = \frac{\sum_{\tau=1}^T \beta_\tau}{T} = \mathbb{E}_{\tau \in [T]} [\beta_\tau] \leq 1 \cdot p + (1 - p) \cdot \frac{1}{4},$$

which can be rearranged to $p \geq \frac{1}{10}$ for $\varepsilon \leq \frac{1}{20}$. In other words, we have $\frac{T}{10} \geq \frac{1}{16}n =: k$ many time units $\tau = \{\tau_1, \dots, \tau_k\}$ in which at least $\frac{n}{4}$ many packets are crossing an edge in e_1, \dots, e_n . Let $A(\tau)$ be the event that there is no collision at time τ . Then we can bound the probability of having no collision at all, by just considering the time units in τ :

$$\Pr \left[\bigwedge_{\tau=1}^T A(\tau) \right] \leq \prod_{j=1}^k \Pr[A(\tau_j) \mid A(\tau_1), \dots, A(\tau_{j-1})] \stackrel{(*)}{\leq} \left(\frac{15}{16} \right)^{\frac{n}{8} \cdot k} = \left(\frac{15}{16} \right)^{n^2/128}$$

It remains to justify the inequality (*).

Claim. *For all $j = 1, \dots, k$ one has $\Pr[A(\tau_j) \mid A(\tau_1), \dots, A(\tau_{j-1})] \leq (\frac{15}{16})^{n/8}$.*

By $P_i(\tau)$ we denote the random variable that gives the edge that i traverses at time τ (in case that i is waiting in a node v , let's say that $P_i(\tau) = (v, v)$). Let $E_i := \{P_i(\tau_1), \dots, P_i(\tau_{j-1})\} \cap \{e_1, \dots, e_n\}$ be the critical edges that packet i has visited at $\tau_1, \dots, \tau_{j-1}$. It suffices to show that $\Pr[A(\tau_j) \mid E_1, \dots, E_n] \leq (\frac{15}{16})^{n/16}$, i.e. we condition on those edges E_i . Let $I \subseteq [n]$ with $|I| = \frac{n}{4}$ be the indices of packets that cross a critical edge at time τ_j . We split I into equally sized parts $I = I_1 \dot{\cup} I_2$, i.e. $|I_1| = |I_2| = \frac{n}{8}$. Consider the critical edges $E^* := \{P_i(\tau_j) \mid i \in I_1\}$ which are chosen by packets in I_1 . If $|E^*| < \frac{n}{8}$ then we have a collision, so condition on the event that $|E^*| = \frac{n}{8}$. Now for all other packets $i \in I_2$, the edge $P_i(\tau_j)$ is a uniform random choice from $\{e_1, \dots, e_n\} \setminus E_i$. Thus we have independently for all $i \in I_2$,

$$\Pr[P_i(\tau_j) \in E^*] = \frac{|E^* \setminus E_i|}{|\{e_1, \dots, e_n\} \setminus E_i|} \geq \frac{n/8 - n/16}{n} = \frac{1}{16},$$

since $|E_i| \leq k = \frac{n}{16}$. Thus

$$\Pr \left[A(\tau_j) \mid |E^*| = \frac{n}{8}; E_1, \dots, E_n \right] \leq \Pr \left[\bigwedge_{i \in I_2} P_i(\tau_j) \notin E^* \mid |E^*| = \frac{n}{8}; E_1, \dots, E_n \right] \leq \left(\frac{15}{16} \right)^{n/8}$$

and the claim follows. \square

Finally one can check that for $\varepsilon := 0.000032$ and n large enough one has

$$\left(\frac{15}{16}\right)^{n^2/128} \cdot 2^{(\Phi(\varepsilon)+o(1))n^2} < 1$$

and Theorem 6 follows.

Observe that in our instance, C and D are within a factor of 2 of each other. In contrast, if $C \gg D$, then there is a schedule of length $(1 + o(1)) \cdot C$ and buffer size $O(\frac{C}{D})$, see [Sch98, Chapter 6].

Acknowledgements. The author is very grateful to Rico Zenklusen for carefully reading a preliminary draft.

References

- [AS08] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., Hoboken, NJ, third edition, 2008. With an appendix on the life and work of Paul Erdős.
- [CI96] A. E. F. Clementi and M. Di Ianni. On the hardness of approximating optimum schedule problems in store and forward networks. *IEEE/ACM Trans. Netw.*, 4(2):272–280, 1996.
- [DP09] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, Cambridge, 2009.
- [EL75] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. North-Holland, Amsterdam, 1975.
- [FS02] U. Feige and C. Scheideler. Improved bounds for acyclic job shop scheduling. *Combinatorica*, 22(3):361–399, 2002.
- [KPSW09] R. Koch, B. Peis, M. Skutella, and A. Wiese. Real-time message routing and scheduling. In *APPROX-RANDOM*, pages 217–230, 2009.
- [LMR94] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-shop scheduling in $O(\text{congestion} + \text{dilation})$ steps. *Combinatorica*, 14(2):167–186, 1994.
- [LMR99] T. Leighton, B. Maggs, and A. W. Richa. Fast algorithms for finding $O(\text{congestion} + \text{dilation})$ packet routing schedules. *Combinatorica*, 19(3):375–401, 1999.
- [MS11] M. Mastrolilli and O. Svensson. Hardness of approximating flow and job shop scheduling problems. *J. ACM*, 58(5):20, 2011.
- [MT10] R. A. Moser and G. Tardos. A constructive proof of the general lovász local lemma. *J. ACM*, 57(2), 2010.
- [MU05] M. Mitzenmacher and E. Upfal. *Probability and computing*. Cambridge University Press, Cambridge, 2005. Randomized algorithms and probabilistic analysis.

- [MV99] F. Meyer auf der Heide and B. Vöcking. Shortest-path routing in arbitrary networks. *J. Algorithms*, 31(1):105–131, 1999.
- [OR97] R. Ostrovsky and Y. Rabani. Universal $o(\text{congestion} + \text{dilation} + \log^{1+\epsilon} n)$ local control packet switching algorithms. In *STOC*, pages 644–653, 1997.
- [PSW09] B. Peis, M. Skutella, and A. Wiese. Packet routing: Complexity and algorithms. In *WAOA*, pages 217–228, 2009.
- [PW11] B. Peis and A. Wiese. Universal packet routing with arbitrary bandwidths and transit times. In *IPCO*, pages 362–375, 2011.
- [RT96] Y. Rabani and É. Tardos. Distributed packet switching in arbitrary networks. In *STOC*, pages 366–375, 1996.
- [Sch98] C. Scheideler. *Universal Routing Strategies for Interconnection Networks*, volume 1390 of *Lecture Notes in Computer Science*. Springer, 1998.
- [ST00] A. Srinivasan and C. Teo. A constant-factor approximation algorithm for packet routing and balancing local vs. global criteria. *SIAM J. Comput.*, 30(6):2051–2068, 2000.
- [Wie11] A. Wiese. *Packet Routing and Scheduling*. Dissertation, TU Berlin, 2011.